Universidade Federal do Ceará – IUFC
Centro de Ciências – CC
Departamento de Computação - DC
Fundamentos de Programação

Exercício: Funções
Objetivos: Introduzir o conceito de funções.
Data da Entrega: 06/06/2022

NOME: _____MATRÍCULA: _____

QUESTÃO 1

Para cada um dos problemas a seguir, elabore um algoritmo utilizando Português Estruturado (Portugol) e, em seguida, implemente o algoritmo concebido utilizando a Linguagem C (para alunos do curso de Engenharia de Computação) ou Python (para alunos do curso de Ciência da Computação).

1.1. Escreva um algoritmo para ler um número natural N e calcular o maior número primo menor do que N. Utilize uma sub-rotina para verificar se um número qualquer é ou não primo.

1.2. Escreva uma sub-rotina float calcula(float x, float y, char op), que realiza um cálculo com as variáveis x e y dependendo do valor de op, segundo os seguintes critérios:

- Se op for igual a ‘a’, os valores de x e y devem ser somados;
- Se op for igual a ‘s’ os valores de x e y devem ser subtraídos;
- Se op for igual a ‘m’ os valores de x e y devem ser multiplicados;
- Se op for igual a ‘d’, os valores de x e y devem ser divididos.

Realizar o teste da sub-rotina float calcula(float x, float y, char op) no programa principal.

1.3. Escreva uma sub-rotina função int converteTempo(int numHoras, char tipo), que converte o número de horas passado pelo parâmetro numHoras na unidade de tempo determinada pelo parâmetro tipo, onde: ‘h’ = horas, ‘m’ = minutos e ‘s’ = segundos.

Exemplos:
converteTempo(5,’h’) = 5
converteTempo(2,’m’) = 120
converteTempo(3,’s’) = 10800

Realizar o teste da sub-rotina int converteTempo(int numHoras, char tipo) no programa principal.

1.4. Escreva uma sub-rotina int eVogal(char letra) que indica se a letra passada como parâmetro da função é ou não uma vogal. A função deve retornar 1, caso a letra seja uma vogal e 0, caso contrário. Elabore um programa para testar a função int eVogal(char letra).

1.5. Escreva uma sub-rotina int fatorial(int x), que deve retornar o fatorial de x.

1.6. Escreva uma sub-rotina void triangulo(int a, int b, int c), que recebe como parâmetros três lados de um triângulo e, caso estes três lados formem um triângulo, mostra o tipo do mesmo. Para que seja possível construir um triângulo, nenhum dos lados pode ser maior ou igual a soma dos outros dois.
O procedimento deve exibir as seguintes saídas:

- "equilatero", caso os três lados sejam iguais;
- "isosceles", caso dois lados sejam iguais;
- "escaleno", casos os três lados sejam diferentes;
- "nao forma triangulo", caso não seja possível formar um triângulo.

1.7. Escreva uma sub-rotina que receba, por parâmetro, dois valores X e Z, calcule e retorne X^Z (sem utilizar funções ou operadores de potência prontos). O programa principal deverá ler as entradas e imprimir os resultados.

1.8. Escreva uma sub-rotina que receba um número inteiro n como parâmetro e retorne o número de divisores de n. O programa principal deverá ler uma sequência de números, terminada pelo flag zero, e calcular o número de divisores de cada um deles.

1.9. Escreva uma sub-rotina que recebe um número unsigned int como parâmetro e retorna este número escrito ao contrário.

Ex: 431 <-> 134.

1.10. Escreva uma sub-rotina que arredonda um valor dado. O número deve ser arredondado para o inteiro mais próximo. Se o número for equidistante de dois inteiros, deve ser arredondado para o valor de maior magnitude. Ou seja, 1.5 é arredondado para 2, e -1.5 é arredondado para -2. O protótipo da função deve ser:

int arredonda (double x);

1.11. Implementar a função cujo cabeçalho é apresentado a seguir:

unsigned char calculaParidade (unsigned char b);

Interferências, ruídos e outros fenômenos que prejudicam a integridade dos dados são problemas fundamentais quando computadores se comunicam em rede. Para detectar alterações em bits, os dados são sempre enviados com redundâncias computadas a partir dos bits originais. Este tipo de técnica de detecção de erros costuma receber o nome de checksum, e segue o mesmo princípio dos dígitos verificadores presentes em diversos documentos e identificadores numéricos (por exemplo, números de contas e agências bancárias). Uma das técnicas de detecção de erros mais simples e mais usadas é o teste de paridade. Cada byte é enviado junto com um bit adicional, que indica se o número de bits com valor 1 no byte é par (bit redundante = 0) ou ímpar (bit redundante = 1). Por exemplo um byte com o valor 8 tem os bits 00001000, ou seja, apenas 1 bit "setado",

portanto a sua paridade é 1. Já um byte com o valor 0x55 é representado pelos bits 01010101 – 4 bits "setados", portanto a sua paridade é 0.

A função calculaParidade deve receber como parâmetro um byte e retornar o valor do bit redundante (0 ou 1).

1.12. Escreva uma sub-rotina que, dado um número real passado como parâmetro, retorne a parte inteira e a parte fracionária desse número por referência.

1.13. Escreva uma sub-rotina que troca o conteúdo de duas variáveis.

1.14. Escreva uma sub-rotina que recebe como parâmetros um vetor real A com n elementos e um vetor real B com m elementos, ambos representando conjuntos, e verifica se A está contido em B.

1.15. Escreva uma sub-rotina que recebe uma matriz real A_{mxn}, o valor de m, o valor de n e retorna o valor 1, se A for uma matriz Identidade, e 0, caso contrário.

1.16. Escreva uma sub-rotina que recebe um vetor e, em seguida, identifica o maior elemento do vetor e armazena esse elemento na primeira posição do vetor.

1.17. Escreva uma sub-rotina que localiza uma letra em um vetor de caracteres (recebidos como parâmetros) e retorna um outro vetor com suas posições onde a letra foi encontrada.

Por exemplo:

0 1 2 3 4
[v][a][n][i][a]

// output procurando a letra "a"
[1][4]

1.18. Escreva uma sub-rotina que recebe dois vetores A e B de tamanho n (contendo exatamente n/2 valores pares e n/2 valores ímpares, respectivamente) e, armazena todos os valores pares em A e todos os valores ímpares em B.

1.19. Escreva uma sub-rotina que recebe um vetor A e inverte a posição de seus elementos. Por exemplo, o primeiro elemento passa a ser o último, o segundo elemento passa a ser o penúltimo e assim sucessivamente. Não utilize vetores auxiliares. Imprima o vetor invertido ao final.

1.20. Escreva uma sub-rotina que recebe um vetor A e um número n, e, em seguida, exclui do vetor A todas as ocorrências do elemento n.

"Quando a educação não é libertadora, o sonho do oprimido é ser o opressor."
Paulo Freire